WHAT IS CLAIMED IS

1. A programmable device configured to implement a finite state machine that may sequence through a plurality of states responsive to a plurality of input signals during a single cycle of a clock for the programmable device, the input signals being arranged from a first signal to a last signal, the programmable device comprising:

a plurality of programmable blocks, each programmable block being instantiated to form a memory, the memories corresponding on a one-to-one basis with the plurality of input signals such that a first memory corresponds to the first input signal, and so on, each memory being programmed to retrieve a data word that represents a next state of the finite state machine, the data word being retrieved according to an address derived both from a current state of the finite state machine and from the corresponding input signal.

2. The programmable device of claim 1, wherein each programmable block is a look-up-table-based programmable block.

3. The programmable device of claim 1, wherein the first through the last input signal correspond to sequential samples of an external bus supporting a wire-line-level protocol.

4. The programmable device of claim 3, wherein the wire-line-level protocol is SPI4.

5. The programmable device of claim 4, wherein each sample of the external bus is sixteen-bits wide.

6. The programmable device of claim 1, wherein the subset of memories are arranged such that the address for a second memory is derived from the retrieved data word form the first memory, an address for a third memory is derived from the retrieved data word from the second memory, and so on such that an address for a last memory is derived from the retrieved data word from a next-to-last memory, and wherein the programmable block instantiating the last memory is configured to register its retrieved data word responsive to the clock, the first memory being arranged to derive its address from the registered data word from the last memory.

7. The programmable device of claim 6, wherein the retrieved data words from the first through the next-to-last memory are non-clocked data signals.

8. The programmable device of claim 6, wherein the subset of programmable blocks are arranged such that the programmable block instantiating the second memory is adjacent to the programmable block instantiating the first memory, the programmable block instantiating the third memory is adjacent to the programmable block instantiating the second memory, and so on.

9. The programmable device of claim 6, wherein the plurality of programmable blocks are arranged in rows and columns and the subset of programmable blocks are arranged in a single column adjacent to one another.

10. The programmable device of claim 6, wherein the plurality of programmable blocks are arranged in rows and columns and the subset of programmable blocks are arranged in a single row adjacent to one another.

11. The programmable device of claim 1, wherein the input signals are arranged from a first to a fourth input signal.

12. The programmable device of claim 1, wherein the input signals are arranged from a first to a fifth input signal.

13. The programmable device of claim 1, wherein the memories are read-only memories.

14. A method of sequencing a finite state machine implemented in a programmable device through multiple states in a single cycle of an internal clock for the programmable device, the programmable device having a set of programmable blocks, the method comprising:

programming each programmable block to instantiate a memory, wherein each memory is programmed to retrieve a data word representing a next state of the finite state machine based upon address signals, and wherein the memories are arranged from a first memory to a last memory;

during each of a sequence of the clock cycles:

registering a plurality of input signals, wherein the plurality of input signals corresponds on a one-to-one basis with the plurality of memories such the plurality of input signals are arranged from a first input signal to a last input signal;

sequencing the finite state machine by sequentially retrieving a first data word form the first memory based upon an address derived both from an initial state and from the first input signal; then retrieving a second data word from the second memory based upon an address derived both from the first data word and from the second input signal; and so on until a last data word is retrieved from the last memory based upon an address derived from both a next-to-last data word and from the last input signal; and

registering the last data word so it may be used to form the initial state for the next clock cycle.

15. The method of claim 14, wherein the plurality of input signals is formed by sequentially sampling an external bus supporting a wire-line-level protocol.

16. The method of claim 15, wherein the wire-line-level protocol is SPI4.

17. The method of claim 14, wherein each memory has a depth of 32 words.

18. The method of claim 14, wherein each data word is 4 bits wide.

19. The method of claim 14, wherein each memory is a read-only memory.

20. A programmable device configured to implement a finite state machine that may sequence through a plurality of states responsive to a plurality of input signals during a single cycle of a clock for the programmable device, the input signals being arranged from a first signal to a last signal, the programmable device comprising:

a plurality of LUT-table based programmable blocks, each programmable block being instantiated to form a ROM , the memories corresponding on a one-to-one basis with the plurality of input signals such that a first memory corresponds to the first input signal, and so on, each memory being programmed to retrieve a data word that represents a next state of the finite state machine, the data word being retrieved according to an address derived both from a current state of the finite state machine and from the corresponding input signal, wherein for an nth memory, n being a positive integer, the current state corresponds to the data word retrieved from the (n-1)th memory.